

CHAPTER 5.

ACCOUNTING FOR INTERNAL USE SOFTWARE

SECTIONS

PURPOSE.....	01
SCOPE	02
AUTHORITY.....	03
POLICY	04
EFFECT ON OTHER ISSUANCES.....	05
EXHIBIT 1: Flow Chart/Decision Tree for the Capitalization of Software	06

05-01 PURPOSE.

This chapter prescribes the policy for determining and recording the value of internal use software. It provides NOAA with the criteria to determine the cost of internal use software (IUS), whether it should be capitalized and recorded in NOAAs property records as an asset, or charged as an expense.

05-02 SCOPE.

This policy is applicable to all components of NOAA and cross-serviced agencies.

05-03 AUTHORITY.

Statement of Federal Financial Accounting Standards No. 10, *Accounting for Internal Use Software* (<http://www.fasab.gov/pdffiles/fasab10.pdf>).

Federal Financial Accounting Standards Advisory Board, *Implementation Guidance for Internal Use Software*; Federal Financial Accounting Technical Release 16, January 19, 2016
http://www.fasab.gov/pdffiles/original_tr_16.pdf

U. S. Department of Commerce, Accounting Principles and Standards Handbook, *Chapter 8, Assets*, http://www.osec.doc.gov/ofm/Accounting/Final_Chapter_8_Acctg_Hdbk_6-3-16.pdf

05-04 POLICY.

05-04.01 INTRODUCTION

The policy applies to software used to perform NOAAs mission and includes both financial and administrative software, including that used for project management, and software used to

produce goods and services. It requires that internal use software be capitalized when the cost to acquire or develop the software is \$200,000 or greater and has an expected useful life of two (2) or more years. This policy encompasses the following types of software:

- Commercial off-the-shelf (COTS) software: COTS software refers to software that is purchased from a vendor and is ready for use with little or no changes.
- Developed software:
 - a. Internally developed software is software that employees of the NOAA develop, including new software and existing or purchased software that is being modified with or without a contractor's assistance.
 - b. Contractor-developed software refers to software that NOAA pays a contractor to design, program, install, and implement, including new software and the modification of existing or purchased software.

05-04.02 SOFTWARE DEVELOPMENT STAGES

Developed software phases include planning/design, development, and operations. The phases of internal use software development help to isolate the capitalization period for internal use software.

Preliminary Planning/Design Phase

- Make strategic decisions to allocate resources between alternative projects at a given time. For example, should programmers develop new software or direct their efforts toward correcting problems in existing software?
- Determine performance requirements (i.e., what is it that they need the software to do?).
- Invite vendors to perform demonstrations of how their software will fulfill a Federal entity's needs.
- Explore alternative means of achieving specified performance requirements. For example, should the Federal entity make or buy the software? Should the software run on a mainframe or a client server system?
- Determine that the technology needed to achieve performance requirements exists.
- Select a vendor if a Federal entity chooses to obtain COTS software.

- Select a consultant to assist in the software's development or installation.

Software Development Phase

- Use a system to manage the project.
- Track and accumulate life-cycle cost and compare it with performance indicators.
- Determine the reasons for any deviations from the performance plan and take corrective action.
- Test the deliverables to verify that they meet the specifications.

Post-implementation/Operational Phase

- Operate the software, undertake preventive maintenance, and provide ongoing training for users.
- Convert data from the old to the new system.
- Undertake post-implementation review comparing asset usage with the original plan.
- Track and accumulate life-cycle cost and compare it with the original plan.

05-04.03 CAPITALIZATION

The cost of internal use software should be capitalized when such software meets NOAAs criteria for capitalizable personal property threshold of \$200,000 and has a useful life of two years or more.

Capitalization of Commercial off the Shelf (COTS) Software

Commercial off-the-shelf (COTS) software is software that is purchased from a vendor and is ready for use with little or no changes. The cost of COTS software recorded in NOAAs Property System must include the amount paid to the vendor for the software. Material internal cost incurred by the NOAA to implement the COTS software or otherwise make it ready for use should be capitalized. If the costs of extra services, such as training, are included with the acquisition, these costs should be expensed.

COTS software requiring material internal cost to implement or make it ready for use will be considered developed software, and all direct and indirect costs, including staff time, contract labor, supplies, materials (including software), rent, benefits, etc., incurred to develop the

software, must be tracked in construction-work-in-progress (CWIP). See section on capitalization of developed software below.

Capitalization of Developed Software:

Only costs incurred during the software development phase are capitalized. Software development is considered complete and costs are no longer accumulated for capitalization when the software passes acceptance testing. All developed software, which will be capitalized, must follow the NOAA CWIP procedures (see section on CWIP below).

- Internally Developed Software
 - a. The cost of internally developed software is the full cost (both direct and indirect costs) incurred during the software development phase. The direct and indirect costs include salaries of programmers, systems analysts, project managers and administrative personnel, associated employee benefits, outside consultants' fees, rent, supplies, and documentation manuals. This may include programming, management, or significant review.
 - b. Such cost should be limited to cost incurred after:
 - Management authorizes and commits to a computer software project and believes that it is more likely than not that the project will be completed and the software will be used to perform the intended function with an estimated service life of 2 years or more, and
 - Completion of conceptual formulation, design, and testing of possible software project alternatives (the preliminary design stage).
- Contractor Developed Software
 - a. The cost of contractor-developed software is the amount paid to a contractor to design, program, install, and implement the software. It should also include any material internal cost incurred by NOAA to implement the software or otherwise make it ready for use. The cost excludes NOAA planning and operational costs.

Costs incurred after final acceptance testing has been successfully completed should be expensed. Where the software is to be installed at multiple sites, capitalization should cease at each site after testing is complete at that site. See Statement of Federal Financial Accounting Standards No. 10, Accounting for Internal Use Software at

http://www.fasab.gov/pdffiles/handbook_sffas_10.pdf for further details on capitalization.

Capitalization of Separate Modules

Capitalization standards may be applied to the total cost of the software or, when appropriate, to individual components or modules. For example, if software consists of more than one component or module, and one module is implemented more than six months before another, the modules will be accounted for and tracked separately, and these policies and procedures will apply to the individual modules rather than the entire software package.

Cost Estimation

Considering economic feasibility, a cost estimation technique could be developed to trace the costs to outputs based on the Statement of Federal Financial Accounting Standards (SFFAS) 4: *Managerial Cost Accounting Standards and Concept*, paragraph 124, provision that “[in] principle, costs should be assigned to outputs in one of the methods listed below in the order of preference:

- Directly tracing costs wherever economically feasible;
- Assigning costs on a cause-and-effect basis; and
- Allocating costs on a reasonable and consistent basis.”

A specific software development project may include expenditures for improvements and maintenance that cannot be easily separated but may be reasonably and consistently allocated. One approach that can be used is a ratio based on the projected work hours for development phase activities relative to other types of work. Such a ratio can be applied to determine the expenditures that should be capitalized. The basis for allocating costs should be consistent with applicable standards and defensible.

For more information on SFFAS 4, see http://files.fasab.gov/pdf/files/handbook_sffas_4.pdf .

Software Development Practices

- Linear/Waterfall Development Method
 - A sequential design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the software development phases.
- Agile Software Development Method
 - A group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. In

an agile project, working software is deployed in iterations of typically one to eight weeks in duration, each of which provides a segment of functionality. Initial planning regarding cost, scope, and timing is usually conducted at a high level, and the project status is primarily evaluated based on software demonstrations.

- Spiral Software Development Method
 - The spiral methodology projects are separated into phases like the waterfall method: planning, risk analysis, engineering, and evaluation. However, they are broken up into incremental releases of the product, or incremental refinement through each time around the spiral and through continuously analysing the requirements and improving the definition and implementation. At each iteration around the cycle, the project is improved and extended.

Refer to FASAB Federal Financial Accounting Technical Release 16 for more information on software development practices at http://www.fasab.gov/pdf/origins/original_tr_16.pdf .

Useful Life of Capitalized Software

Capitalized software will be considered to have a useful life of seven years unless justification is provided to the NOAA Personal Property Management Branch (PPMB) for assigning a lesser or greater useful life.

Acceptance Date for Capitalization

COTS software purchased from a vendor should follow NOAA capitalization policies for purchased assets. The acquisition date is the date when the asset is received, inspected, and accepted by a NOAA representative. Unless the contract specifically states otherwise, title will pass to NOAA upon acceptance, regardless of when or where the government takes physical ownership. If the contractual document contains specific guidelines for the acceptance of the asset, the delivery or constructive delivery shall be based on the terms of the contract regarding shipping and/or delivery, and not based on physical possession. Refer to NOAA's Capitalized Personal Property Policy for more information at:

http://www.pps.noaa.gov/financial_policies/capitalized-personal-property-policy.html .

Software, which is developed and cost \$200,000 or more, should be recorded as CWIP until the asset has been placed in service. Software classified as CWIP is capitalized when final acceptance testing has been successfully completed in a production environment.

For each module or component of a software project, the acceptance date will be the date when that module or component has been tested successfully. If the use of a module is dependent on completion of another module(s) the acceptance date will be the date when both that module and the other module(s) have successfully completed testing.

05-04.04 ENHANCEMENTS

Enhancements to existing internal use software must be carefully considered to determine if they should be treated as an internal use software development activity. Significant enhancements of internal use software that lead to and provide significant additional capabilities should be capitalized. However, minor enhancements (i.e. less than \$200,000) resulting from ongoing systems maintenance should be expensed in the period incurred. Furthermore, the cost incurred solely to repair a design flaw or to perform minor upgrades that may extend the useful life of the software without adding capabilities should be expensed.

Capitalizable enhancements normally require new software specifications and may require a change to all or part of the existing software specifications. Examples of enhancements could include augmenting existing functions with new features and functions, developing additional new functions, and/or adding new functionality and capability.

If one module is dependent upon another to function, then those modules should be evaluated together as one enhancement. All costs of an enhancement, including any costs carried over or allocated from the original software, should be amortized over the enhancement's estimated useful life.

05-04.05 DATA CONVERSION COSTS

All data conversion costs incurred for internally developed, contractor-developed, or COTS software should be **expensed** as incurred, including the cost to develop or obtain software that allows for access or conversion of existing data to the new software. Such cost may include the purging or cleansing of existing data, reconciliation or balancing of data, and the creation of new/additional data.

05-04.06 RESEARCH AND DEVELOPMENT SOFTWARE

Research and development (R&D) refers to costs incurred in support of the search for new or refined knowledge and ideas and for the application or use of such knowledge and ideas for the development of new or improved products and processes with the expectation of maintaining or increasing capacity or yielding other future benefits. All R&D cost incurred for software should be charged as an expense when incurred. The development of R&D software may involve testing that software in practical, operational situations as part of the R&D process, but this testing does not, by itself, imply future operational use. R&D does not include routine or periodic alterations to existing software even though these alternations may represent improvements to the software.

R&D software does not include software used to manage, account for, or report on R&D projects. Such software serves an administrative function and is subject to the definitions and requirements of the internal use software capitalization policy.

05-04.07 CONSTRUCTION-WORK-IN-PROGRESS (CWIP)

CWIP is a temporary holding account used to track costs during the design and development of software to be used internally (personal property). In order to be considered a CWIP activity, a software development project must meet all of the following four criteria for capitalization:

- Have an aggregate acquisition cost of \$200,000 or more, and
- Have an estimated service life of two years or more, and
- Provide a long-term future economic benefit to the NOAA organization which maintains or obtains control, and
- Is not intended for sale in the ordinary course of operations.

The CWIP policy must be followed even if a software development project is expected to begin and finish in the same fiscal year (FY). More information on NOAA's CWIP Policy can be found at <http://www.corporateservices.noaa.gov/finance/CWIP%20Policy%20and%20Appendix.html>.

05-04.08 IMPAIRED SOFTWARE

Impairment should be recognized and measured when one of the following occurs and is related to post-implementation/operational software and/or modules:

- The software is no longer expected to provide substantive service potential and will be removed from service, or
- A significant reduction occurs in the capabilities, functions, or uses of the software (or a module thereof).

If the impaired capitalized software is to remain in use, the loss due to impairment should be measured as the difference between the book value and either (1) the cost to acquire software that would perform similar remaining functions (i.e., the unimpaired functions) or, if that is not feasible, (2) the portion of book value attributable to the remaining functional elements of the software. The loss should be recognized upon impairment, and the book value of the asset reduced accordingly. If neither (1) nor (2) above can be determined, the book value should continue to be amortized over the remaining useful life of the software.

If the impaired capitalized software is to be removed from use, the loss due to impairment should be measured as the difference between the book value and the net realizable value (NRV), if any. The loss should be recognized upon impairment, and the book value of the asset reduced accordingly. The NRV, if any, should be transferred to an appropriate asset account until the software is disposed of and the amount is realized.

05.04.09 CLOUD COMPUTING

Cloud computing service is a resource provided over the Internet that has the following essential characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. The most common cloud service resources are software as a service, platform as a service, and infrastructure as a service.

If a cloud computing arrangement includes a software license, the software license element of the arrangement should be accounted for consistent with the acquisition of other software licenses in accordance with the lease criteria stated in SFFAS 5 and SFFAS 6. For more information, refer to SFFAS 5 and SFFAS 6 at <http://www.fasab.gov/document-by-chapter/>.

05.04.10 SHARED SERVICES

Shared Service means a mission or support function provided by one organization within or between organizations. The funding and resourcing of the service is shared and the providing entity effectively becomes an internal/external service provider. There are three types of shared service structures in the Federal government:

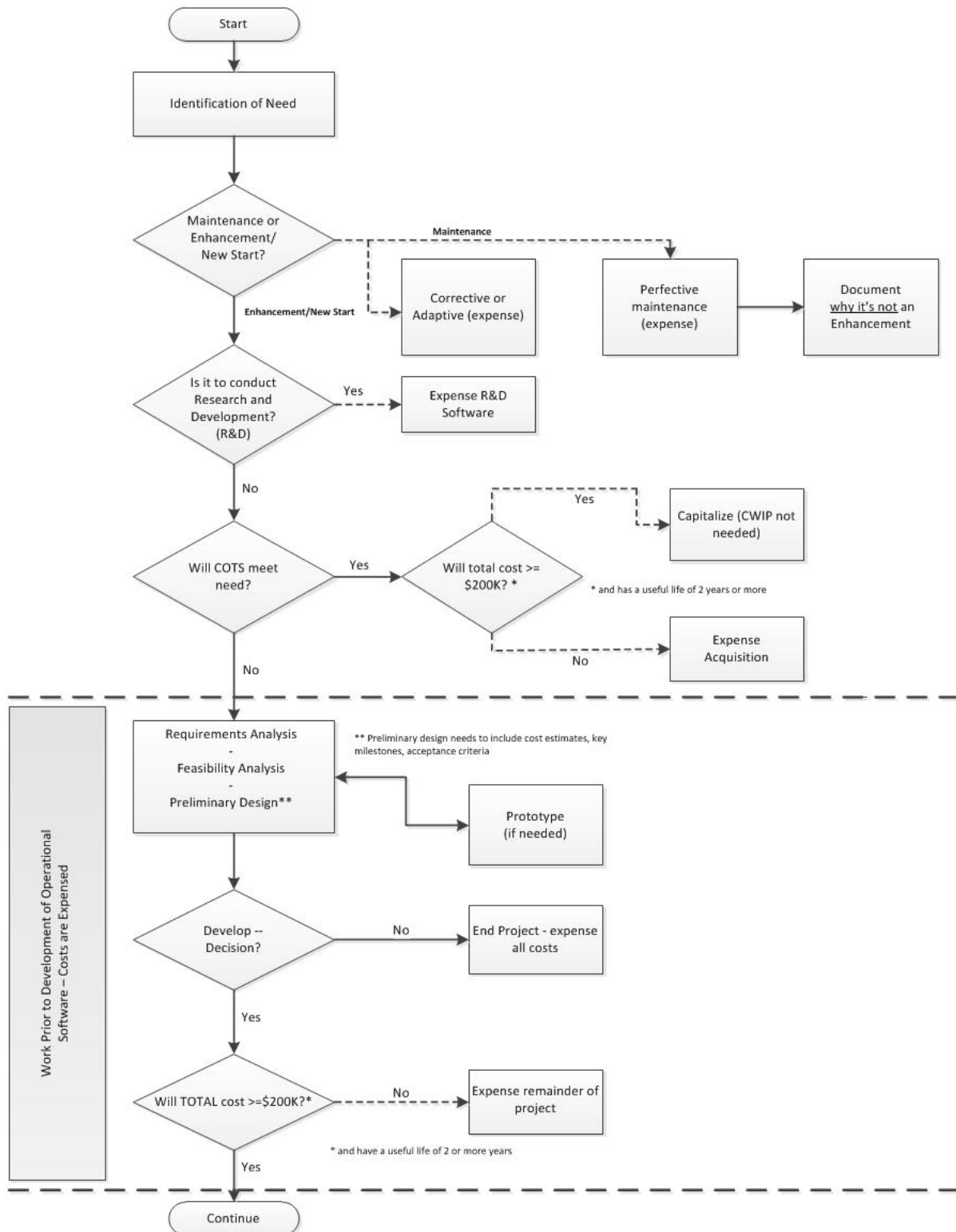
- Intra-agency shared services that include services provided within the boundaries of a specific organization such as a Federal department or agency, to that organization's internal units (i.e. between NOAA and DOC or other DOC bureaus).
 - For intra-agency shared services, a cost allocation methodology could be developed in accordance with SFFAS 4, paragraphs 120-125.
- Interagency-shared services are those provided by one Federal organization to other Federal organizations that are outside of the provider's organizational boundaries (i.e. between Federal agencies).
 - For interagency-shared services and commercial shared services, the service provider entity that owns (receives funding/responsible for maintaining) the software should account for the software in accordance with SFFAS 10.
- Commercial shared services are those provided by private vendors.
 - If the entity receiving the service (the "customer") has a contractual right to take possession of the software at any time during the hosting period without significant penalty, and it is feasible for the customer to either run the software on its own hardware, or contract with another party unrelated to the vendor to host the software, then the customer should account for the software in accordance with SFFAS 10.

See SFFAS 4 and SFFAS 10 for more information at <http://www.fasab.gov/document-by-chapter/> .

05-05 EFFECT ON OTHER ISSUANCES.

This Chapter supersedes Chapter 5 of the NOAA Finance Handbook, dated January 9, 2015.

5-06 EXHIBIT 1. FLOWCHART/DECISION TREE FOR CAPITALIZATION OF SOFTWARE



5-06 EXHIBIT 1. FLOWCHART/DECISION TREE FOR CAPITALIZATION OF SOFTWARE

